Real-Time Hardware Implementation of Telephone Speech Enhancement Algorithm

A Thesis

Presented in Partial Fulfillment of the Requirements for
the Degree Bachelor of Science with Honors Research Distinction in
Electrical and Computer Engineering

By

Forrest Obnamia

* * * * *

Electrical and Computer Engineering

The Ohio State University

2013

Examination Committee:

Dr. Ashok Krishnamurthy, Primary Adviser

Dr. Lawrence Feth, Secondary Adviser

Dr. Bradley Clymer

ABSTRACT

Hearing impairment detrimentally affects communication over the telephone. Since phone lines reduce bandwidth and dynamic range, the poor quality speech signal can cause hard of hearing (HoH) listeners to experience extreme frustration and inefficient communication. One possible solution has been developed at the Ohio State University to help combat this problem. The Telephone Speech Enhancement Algorithm (TSEA) has been created to improve telephone signals so that speech is more intelligible for HoH listeners. Tests for TSEA have been run on human subjects and proven the algorithm effective. However, a hardware implementation of TSEA has yet to be designed. In this thesis, the BeagleBoard-xM development board is used to run TSEA. The software for TSEA is modified so that it can be implemented on the BeagleBoard-xM and tested in a real-time environment. This hardware model runs TSEA but introduces noise into the system due to its analog nature. The model accepts analog audio signals, processes them using TSEA, and outputs the processed signal for transmission. A device such as this has the potential to improve communication in scenarios such as telemedicine clinics where a failure to communicate properly with their HoH customers could have potentially devastating consequences. Ideally if a commercial model was developed, TSEA could be implemented everywhere to help improve communications for the HoH community. This project is the next step in making it a reality.

To my mother and father and Becky.

ACKNOWLEDGMENTS

I would like to thank Dr. Ashok Krishnamurthy for being my adviser and presenting me the opportunity to work on this project. Thank you to Dr. Lawrence Feth for providing advice and sitting in on the oral defense committee. His expertise helped determine the testing and validation methods used. PhD student Sidharth Balasubramanian was gracious enough to listen to my presentation and provide valuable feedback and suggestions for this project. His suggestions for testing the model aided in gathering results crucial to examining its performance. Also, thank you to Dr. Bradley Clymer for helping with the logistical efforts and for being a part of the defense committee.

It is also right to acknowledge all the previous students and faculty who have helped with the Telephone Speech Enhancement Algorithm project. The work done by Harikrishna Natarajan and Resmi Kommatil provided the groundwork for TSEA that I was able to build off of in this thesis. Without their contributions none of this would be possible.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table **Page**

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation

Hearing impairment detrimentally affects communication over the telephone. Hard of Hearing (HoH) listeners have reduced audibility compared to normal listeners which complicates telephone communication. The complication is further impacted by the characteristics of the telephone line.

An estimated 40-45% of adults above the age of 50 experience hearing loss [1]. This statistic grows to 90% for adults above 80 years of age [2]. Hearing loss of this type creates difficulties for many adults when trying to understand speech, especially in the presence of background noise.

When communicating over the telephone, hearing impairment worsens speech perception. Although the use of hearing aids could assist HoH listeners, it is known that only one in five individuals who could benefit from hearing aids actually wear one [3, 4]. To make matters worse, it is a common complaint that hearing aids are uncomfortable in general and especially during telephone calls. Furthermore, only one in four hearing aid owners use amplified telephones [5]. These statistics alone prove there is a demand for improved telephone communication within the HoH community.

Another limitation for HoH listeners is a total reliance on the auditory signal due to a lack of visual cues. Many HoH individuals rely on visual cues such as lip reading and facial expressions to aid in the perception of speech [6]. Communication via telephone does not allow listeners the benefit of visual cues and forces total reliance on the pure acoustic speech signal which makes speech perception more difficult.

A problem that stems from the physical telephone line itself is a reduced bandwidth. The frequency response of a landline telephone is accepted to be 300 to 3300 Hz but the frequencies important for speech can exist up to 8000 Hz. It is consonant sounds in particular that have information in this frequency range that is needed for proper identification. This reduction in frequency bandwidth leaves the listener without signal information that can be important for speech interpretation.

Another problem with telephone lines is a reduced dynamic range. Telephone systems are approximately linear for only a limited range of signal amplitudes. Signals whose amplitudes are above the upper threshold will get clipped and cause distortion. This property can significantly alter the intelligibility of speech.

1.2 Proposed Solution

An algorithm known as the Telephone Speech Enhancement Algorithm (TSEA) has been developed at The Ohio State University to aid in telephone communication with HoH individuals. Through a collaborative effort by the faculty and students in the Department of Electrical and Computer Engineering and the Department of Speech and Hearing Science, TSEA was designed to preprocess speech signals before transmission over the phone line. This preprocessing technique enhances the signal to provide the listener at the other end with a signal that is better suited for speech perception.

TSEA was originally developed and tested in the MATLAB environment [7]. Further work was done to bring TSEA into a real-time implementation using Simulink [8]. This version of TSEA was also created as an executable program that can be run directly on a PC. Testing was conducted on all versions of TSEA and results have shown improved speech perception by as much as 73% for some individuals [9].

This thesis's objective is to implement TSEA on a hardware interface that can preprocess speech signals. The hardware chosen for this project is the BeagleBoard-xM hardware platform. Ideally, the hardware can sit between an analog audio source (telephone) and audio output port to optimize the signal for speech perception using TSEA. A sample setup is shown in Figure 1.1 below.

Figure 1.1: TSEA hardware processer between an audio source and output port [10]

1.3 Thesis Organization

This thesis describes the research conducted during the 2012-2013 school year on the Telephone Speech Enhancement Algorithm. The work is organized as follows:

- Chapter 2 gives a background on the Telephone Speech Enhancement Algorithm.
- Chapter 3 describes the hardware implementation of TSEA on the BeagleBoard.

- Chapter 4 outlines the testing and results gathered.
- Chapter 5 gives the conclusions and recommended future work for TSEA.

CHAPTER 2

THE PREPROCESSING ALGORITHM

The preprocessing algorithm, also known as the Telephone Speech Enhancement Algorithm (TSEA), is used to process speech signals prior to transmission over the telephone line. TSEA maintains speech intelligibility by compensating for the adverse effects of the telephone and the hearing characteristics of the Hard-of-Hearing (HoH) listener. Originally developed by Natarajan [7], the algorithm uses a multi-channel compression technique.

This chapter reviews the work done on TSEA prior to the work performed in this thesis. It is organized as follows:

- Section 2.1 describes major factors that impact the algorithm.
- Section 2.2 explains how the algorithm works.
- Section 2.3 describes the real-time implementation of TSEA using Simulink.
- Section 2.4 shows the results gathered from testing the algorithm.
- Section 2.5 explains the need for the work done in the current thesis.

2.1 Impacting Factors

The two main factors in speech intelligibility over the telephone for HoH listeners are the characteristics of hearing impairment and nonlinearity of the telephone. These are described in the following subsections.

2.1.1 Hearing Characteristics of HoH Listeners

Sensorineural hearing loss (SNHL) is the most common type of hearing loss accounting for 90% of hearing loss cases [1]. It is known to occur in 23% of the population above the age of 65 [11]. SNHL occurs when there is damage to the inner ear (cochlea or cochlear nerve). This type of hearing loss is the primary focus of TSEA due to its prevalence in the community. There are two main characteristics of SNHL that need to be explained for the sake of understanding TSEA properly: frequency selectivity and abnormal loudness recruitment.

Frequency selectivity is the ability of the auditory system to distinguish and resolve individual frequencies of a complex sound [12]. For people with SNHL, the damage in the cochlea creates a reduced tuning ability on the basilar membrane and in the neurons of the auditory nerve. Sinusoidal components of a complex sound create a peak in the vibration pattern at a point on the basilar membrane within the cochlea. In normal listeners, this peak will be sensed within an auditory filter and processed individually. However, listeners with SNHL have widened auditory filters which cannot separate the sinusoidal components properly. The cochlear damage is what leads to a reduced frequency selectivity and ultimately results in reduced auditory perception.

Abnormal loudness recruitment refers to a property that occurs in SNHL listeners where loudness (a psychoacoustic property based on the brains interpretation of auditory sensation) grows unusually quick as the auditory signal is increased [12]. This phenomenon is best illustrated in Figure 2.1 below.

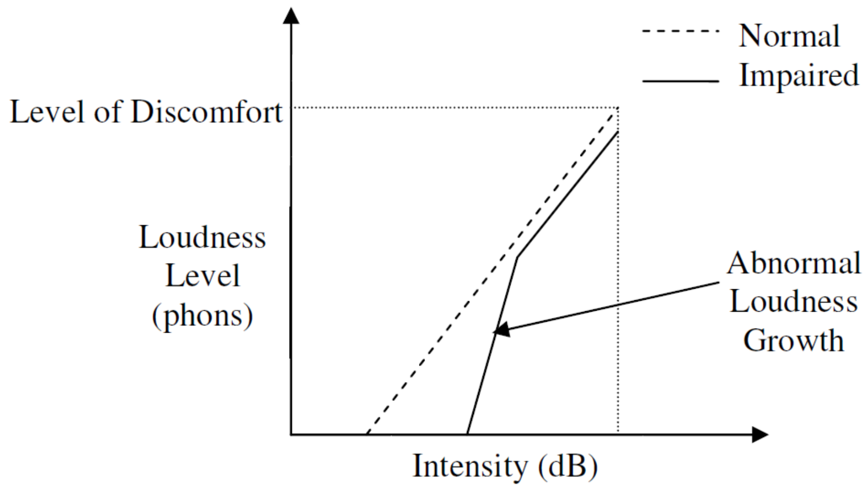

Figure 2.1: Abnormal Loudness Growth in Hearing Impaired Listeners [8]

In Figure 2.1, it can be seen that as a signal's intensity increases (in dB SPL) a normal listener interprets the loudness linearly throughout the audible dynamic range. For a hearing impaired listener, hearing perception does not start at the same threshold of audibility but high-level intensities remain at the same loudness as those for normal listeners. This causes a steep curve known as abnormal loudness growth for the intensity levels between these points. Loud signals that are amplified can cross over the threshold of discomfort of HoH listeners more quickly and cause uncomfortable and painful results. This raised threshold of audibility and same threshold of discomfort in HoH listeners leads to an overall reduced dynamic range.

2.1.2 Telephone Effects on Speech

Telephone lines are analog in nature and have characteristics that impact the signal quality. The two largest factors are described as follows [13]:

- Dynamic Range – The non-linearity of the phone line reduces its dynamic range. The telephone line attenuates signals with amplitudes above a certain range. The clipping of the signals can cause audible distortions to the speech signal that reduce the intelligibility of speech.
- Bandwidth – The telephone line has a limited bandwidth of 300 to 3400 Hz while sound frequencies between 500 and 4000 Hz are most important for speech [11]. The telephone line attenuates signal frequencies outside of its bandwidth which are typically consonants. This phenomenon reduces the intelligibility of speech.

A study was conducted in [7] to determine the telephone line's exact effect on transmitted speech signals. The frequency response of the phone line studied is shown in Figure 2.2 below. It can be seen that the telephone line is most sensitive to middle frequencies and attenuates low and high frequencies. The preprocessing algorithm takes this attenuation into account as described in Section 2.1.3.

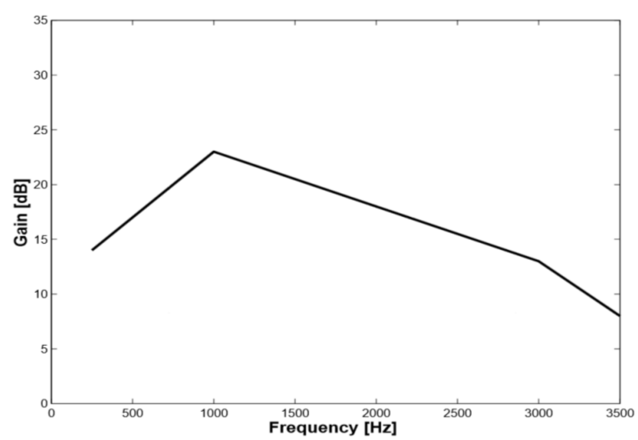

Figure 2.2: Frequency response characteristic of the phone line [7]

2.1.3 Adapted Audiogram Solution

Both the characteristics of hearing impairment and the telephone line need to be taken into account for the algorithm to function optimally. The final solution is an adapted audiogram that can be used within the algorithm. An audiogram is a graph that shows the threshold of audibility over a set of frequencies for a listener. Values from the audiogram are used by the algorithm as described in Section 2.2.2.

The hearing impairment characteristics were taken into account using an "average audiogram" made up from 100 individual audiograms of HoH people over the age of 60 years [9]. These were taken from the Columbus Speech and Hearing Center and provided by the Department of Speech and Hearing Science at The Ohio State University. Figure 2.3 shows average audiogram derived. This figure also shows the reduced dynamic range for HoH listeners when compared to normal listeners.

Telephone characteristics were then applied to the average audiogram to produce an audiogram to be used within the preprocessing algorithm. Figure 2.4 below shows the result. This modified audiogram is the basis for the signal modification performed by TSEA.

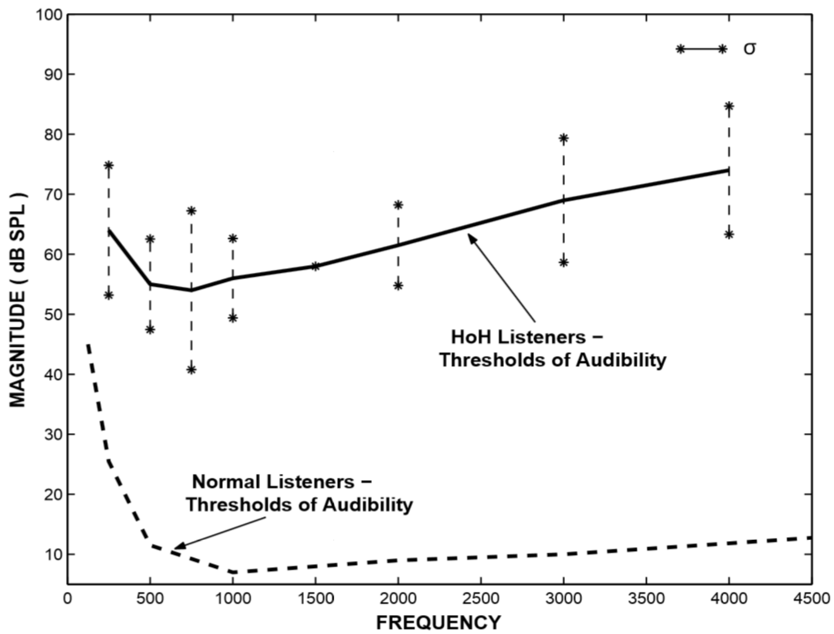

Figure 2.3: Average audiogram for HoH listeners [7]

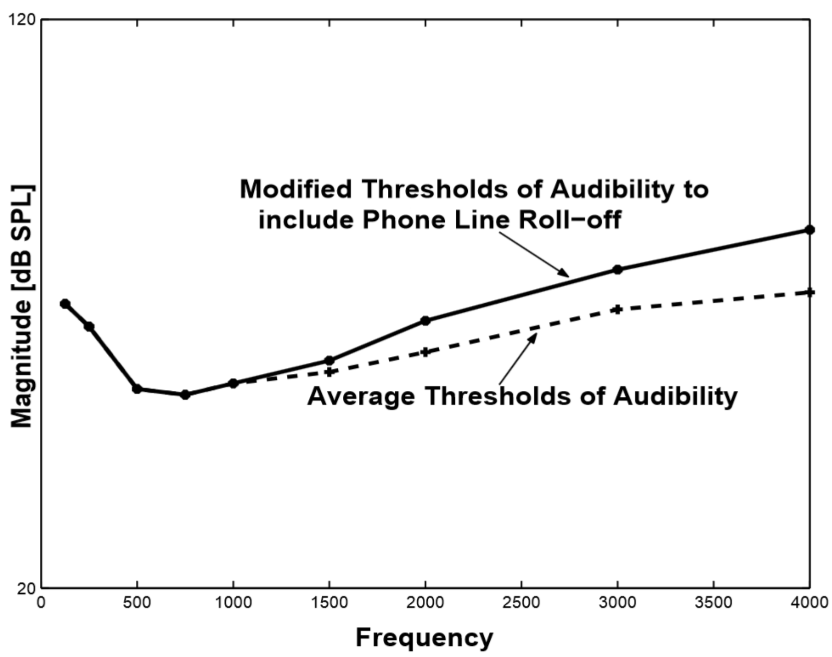

Figure 2.4: Illustration of the adapted audiogram solution [7]

2.2 Algorithm Description

Original development of the preprocessing algorithm was done by Natarajan [7] and is based on compression algorithms investigated by Sheffield et al [14]. The Telephone Speech Enhancement Algorithm (TSEA) takes time domain input speech signals and processes them using a multi-channel compression technique in the frequency domain. After processing, the signal is then returned to the time domain for output.

TSEA's goal is to reduce typical amplitude variations in speech to match the reduced dynamic range of the HoH listener. Ultimately, this will amplify low amplitude consonant sounds above the elevated threshold of audibility of the hearing impaired listener while leaving higher amplitude vowel sounds comfortable loud.

A block diagram of the algorithm can be seen in Figure 2.5. The speech signal is sampled at 8000 Hz and split into 32 msec frames with a 50% overlap between the frames. Each frame is passed through a Hamming window and then a 512-point FFT to compute the spectral content of the frame. If the spectral content is above the noise threshold processing is continued.

For each frame containing a non-noise signal, the spectral content is divided into three channels as described in Section 2.2.1. In order to maximize speech intelligibility, the gain applied must preserve the spectral peak-to-valley ratio as explained in Section 2.2.2. Gain variation between frames is controlled to reduce extreme gain fluctuations as described in Section 2.2.3.

Once spectral processing is complete, the frame is returned to the time domain using an inverse FFT. Frames are then recombined using an over-lap add method to smooth out discontinuities at frame boundaries.

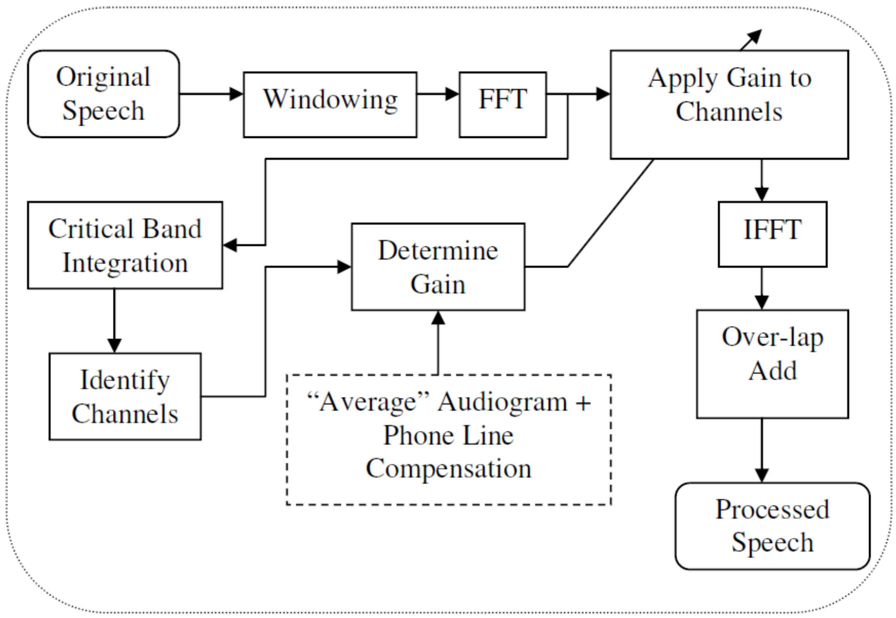

Figure 2.5: Block diagram for the multi-channel compression algorithm (TSEA) [7]

2.2.1 Channel Identification

Once a non-noise frame has been determined, the magnitude spectrum of the frame passes through a "critical band integration" module [7]. This module calculates the average spectrum level in one-third octave frequency bands which is then used to identify major peaks within the frame using a slope-change algorithm. The three largest peaks are chosen so that there is at least one band residing between them. These major peaks correspond to formant frequencies important in distinguishing human vocal sound character. Channel boundaries are then placed equidistant between the peaks at their midpoint bisector to define the channel ranges. After the channels are determined, the average spectrum level is calculated for each channel. Figure 2.6 below shows a sample spectrum after passed through the critical band integration module.

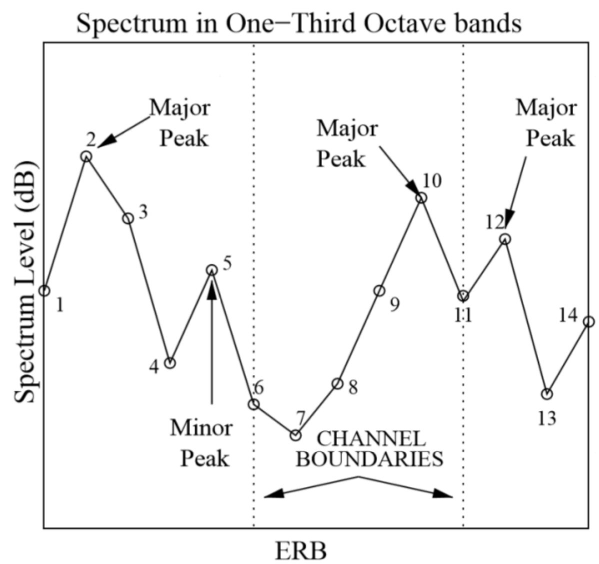

Figure 2.6: Signal spectrum passed through the critical band integration module [7]

2.2.3 Smooth Gains Across Frames

It is important to make sure that the gain does not fluctuate too quickly across frames. In order to solve this problem, attack and release constraints are applied to compress gain variability. When applied, the gain will be constrained within a certain level of the previous frame's gain based on a factor derived from the attack and release values (5msec and 100msec respectively). This leads to smooth gain transitions that prevent fast variations in amplification that cause audible artifacts and reduced speech perceptibility.

2.2.2 Determining Channel Gains

When determining the channel gains, it is important to preserve spectral peak-to-valley ratios in order to maintain intelligibility of speech. Figure 2.7 below shows the method outline for calculating the gain while preserving spectral contrast. The thresholds of audibility used to calculate the gain are taken from the audiogram solution described in Section 2.1.3.

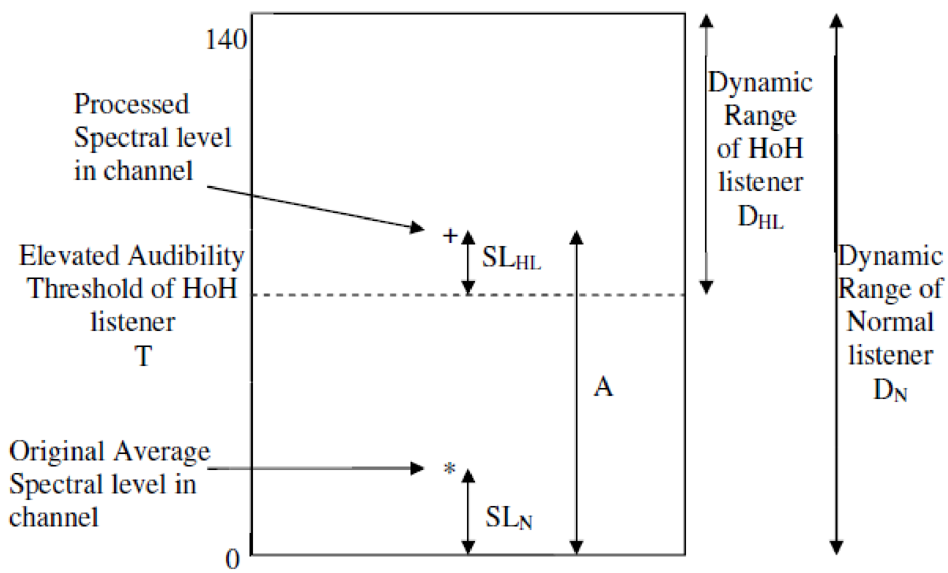

Figure 2.7: Graphical explanation for calculating gain in a channel [8]

Since the objective is to maintain spectral peak-to-valley ratios, the signal gain (compression ratio) G is set to the ratio of D_{HL}, the reduced dynamic range of a HoH listener, and D_N, the dynamic range of a normal listener. Equation 2.1 demonstrates this relationship.

$$G = \frac{D_{HL}}{D_N} \quad [7] \tag{2.1}$$

If T represents the elevated audibility threshold of a HoH listener, the signal must be amplified above the threshold. If SL_N is the spectral level of the unprocessed signal above the threshold of audibility of a normal listener, and A is the spectral level of the processed signal above the same threshold, then the desired A value can be calculated using equation 2.2.

$$A = GSL_N + T \qquad [7] \tag{2.2}$$

This method makes the ratio of SL_{HL}, the spectral level of the processed signal above T, and SL_N equivalent to the ratio G. Thus, the peak-to-valley spectral ratio is preserved.

In order to prevent large gain variations across channel boundaries, the gain is varied linearly across the boundaries to create a smooth transition. This can be seen in Figure 2.8 below. Figure 2.8 also illustrates how the processed spectrum is amplified above the threshold of hearing defined by the adapted audiogram in Section 2.1.3.

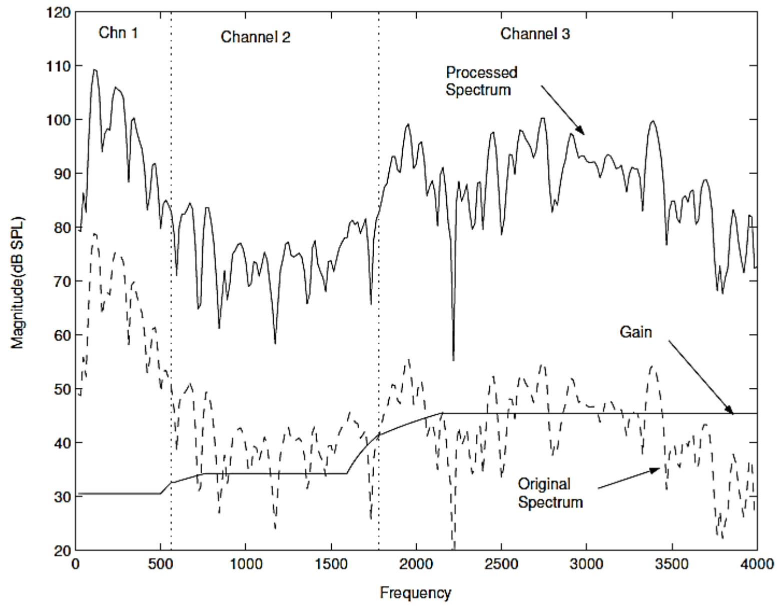

Figure 2.8: Gain preserving spectral contrast with smooth channel transition [8]

2.3 Real-Time Implementation

Further work to TSEA was done by Kommatil [8] and provides the basis for the work conducted in this thesis. TSEA was brought into real-time use by being implemented in a block diagram model using the MathWorks software package Simulink as described in Section 2.3.1. It was then converted to a stand-alone executable program using Real-Time Workshop with an interactive user interface as explained in Section 2.3.2.

2.3.1 Simulink Implementation

Simulink provides users with a graphical user interface (GUI) for designing models as block diagrams. The Signal Processing (DSP) Blockset is a function library that provides all the common blocks used for digital signal processing. These blocks are used to create the model within Simulink which provides the model definition and simulation environment.

In order for the model to accept incoming speech signals, the Microphone (From Wave Device) block was used to capture the signal which it sampled at 8000 Hz with a sample width of 16 bits and 128 samples per frame. Once the signal is being captured, the model processes it according to the specifications laid out in Section 2.2. After the preprocessing is complete, the signal loses its original frame structure and needs to be reconverted back into frames before being output using the Speaker (To Wave Device) block. The processed signal is also written to a WAV file using the To Wave File block for analysis purposes. Figure 2.9 below shows the top level view of the preprocessing algorithm implemented in Simulink.

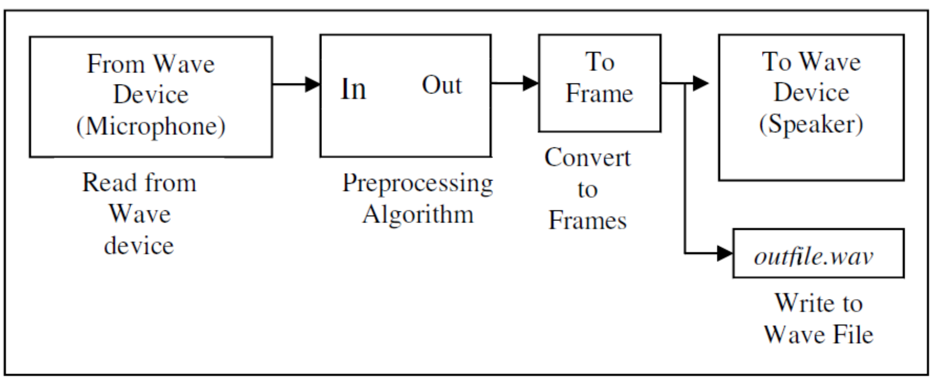

Figure 2.9: Block diagram view of the Simulink implementation [8]

Other versions of the Simulink model were created to aid in testing of the algorithm [8]. These include a version that reads 1-channel (mono) WAV file inputs and a model that reads in 2-channel (stereo) WAV file inputs when channels are identical. Also created is a model that reads 2-channel WAV file inputs when channels are different.

Testing of the model verified that it processes the signal almost identically to the MATLAB version. Figure 2.10 shows the output results of the original waveform and the processed waveforms of both the MATLAB and Simulink design. Visually the processed waveforms are very similar and the RMS error of the two signals is found to be 0.28% with a maximum error between the two outputs of 4.06% [8]. The Simulink model introduces a delay of 128 samples or 16 msec. This is due to the buffer block required to create a 50% overlap of frames. Figure 2.11 shows the delay introduced.

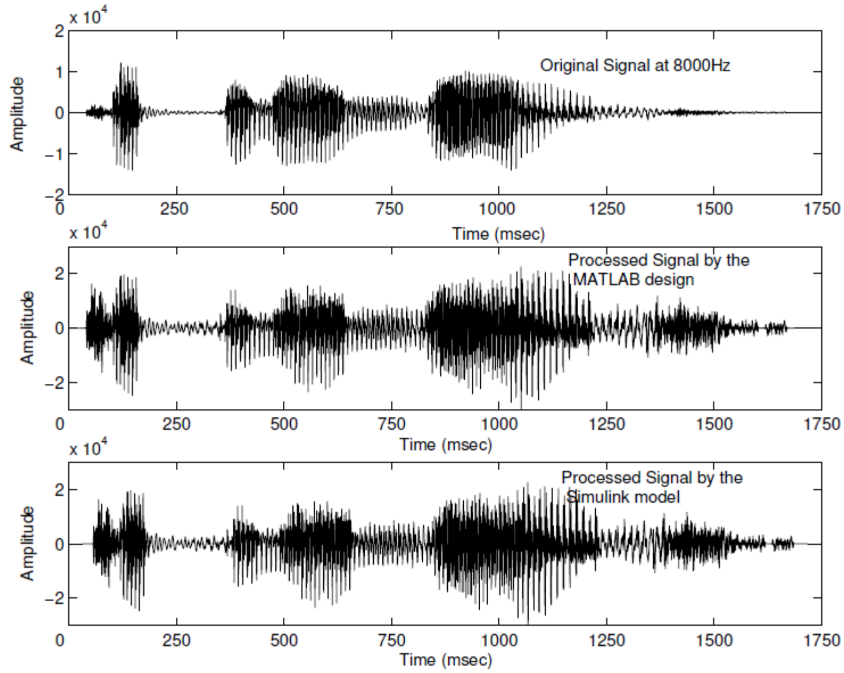

Figure 2.10: The original and processed waveforms [8]

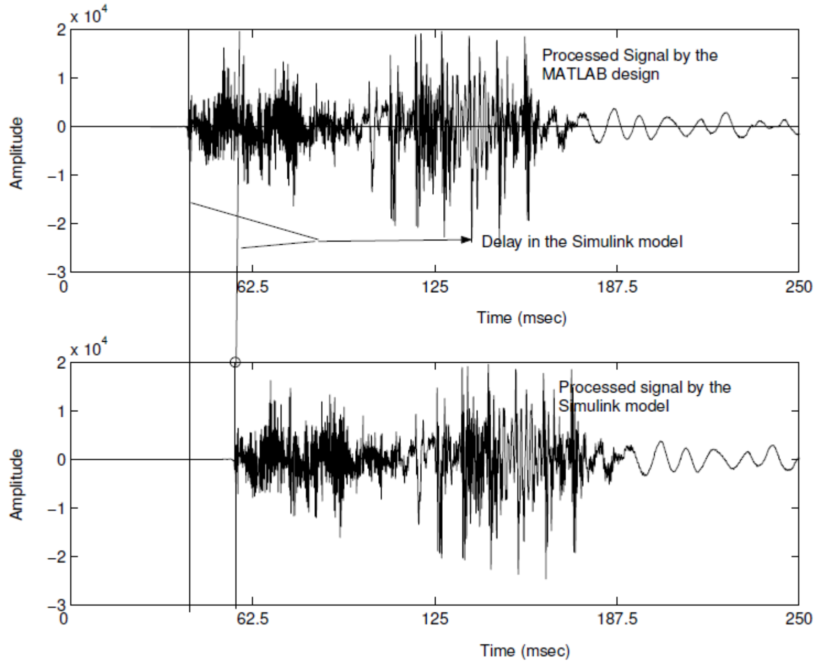

Figure 2.11: Delay of 128 samples (16 msec) introduced by the Simulink model [8]

2.3.2 Stand-Alone Executable Implementation

Real-Time Workshop was used to generate code for a stand-alone executable version of the model that could be run directly on a PC using the command prompt. It takes the Simulink Model and generates stand-alone ANSI C code. The Make utility then takes the ANSI C code and produces the stand-alone executable program. Using the Rapid Simulation (Rsim) Target option in Real-Time Workshop allowed access to parameter tuning without having to recompile the code each time. This provides extra options and flexibility during testing.

A user interface was developed to enable the user to launch the executable program from the interface rather than command prompt. This GUI can be seen in Figure 2.12.

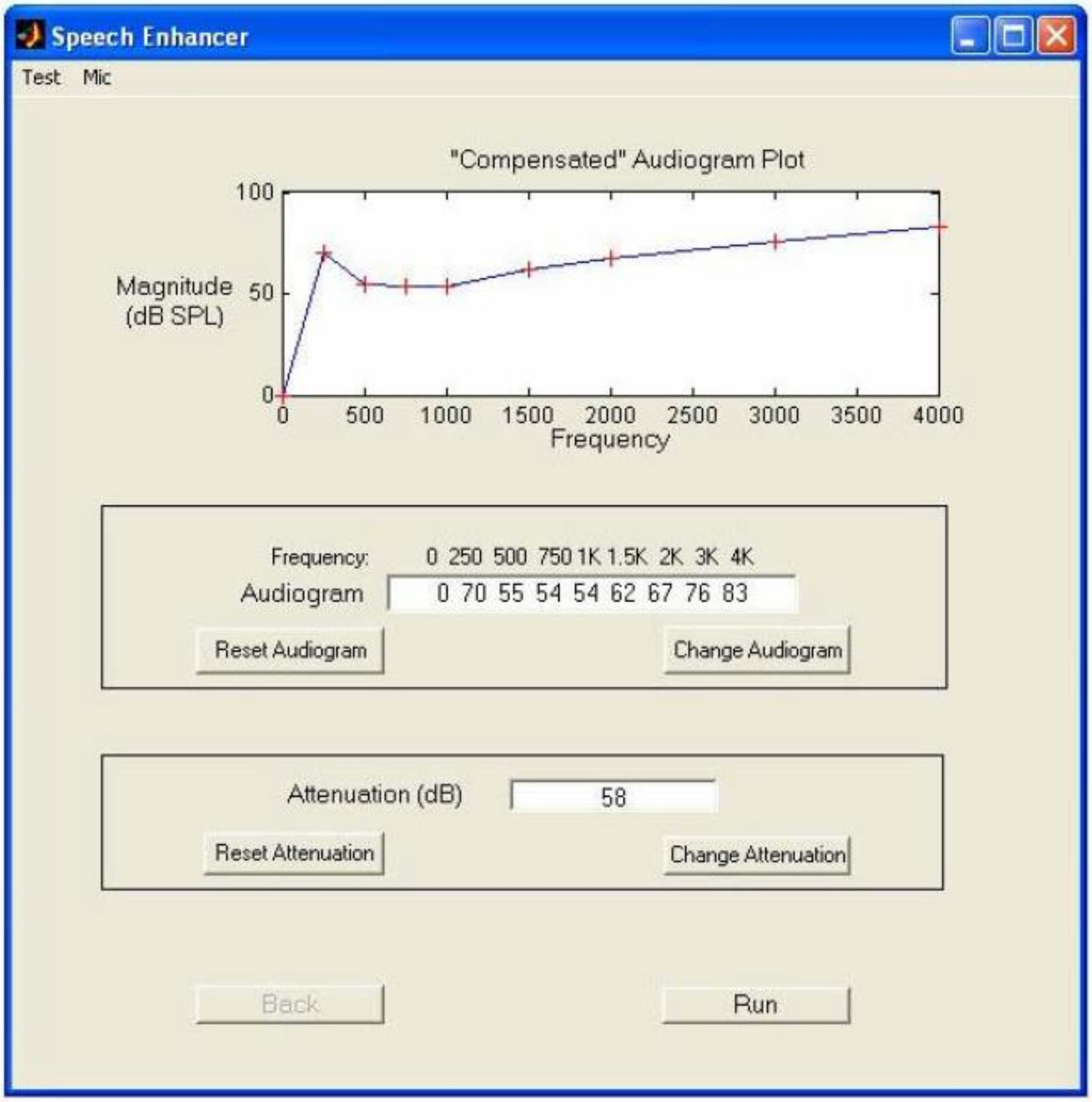

Figure 2.12: Screenshot of the GUI developed for Stand-Alone use [8]

This GUI only runs in the MATLAB environment. It allows the user to select either a microphone input or WAV file input using the Mic and Test menu respectively. It also allows for audiogram and attenuation parameter tuning for increased testing capability and flexibility. Clicking the Run button will start the stand-alone executable program with the parameters selected in the GUI. If the executable is set to read a set of WAV files, the Run button's name will change to the selected filename. Once clicked, the code will run and the name will then change to the WAV file next in the list. The back button is used to repeat the playing of a WAV file that has just been previously played. When Microphone mode is selected, the Run button displays "Run" and the algorithm is run with a microphone input for a configurable amount of time [8].

2.4 Results

Preliminary testing of the algorithm conducted in [7] was performed to determine the advantage of preprocessing the speech signals before transmission over the telephone. Tests were conducted on eight listeners with normal hearing characteristics. Hearing loss was simulated by passing the speech signal through a hearing loss model. Figure 2.13 shows the testing setup used.

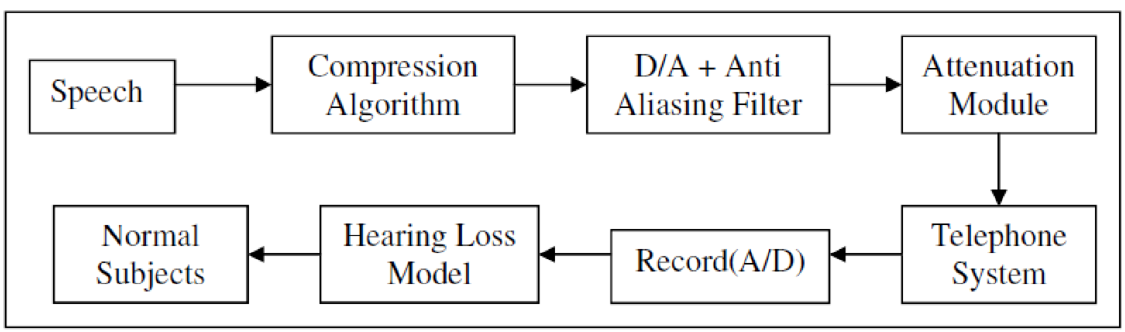

Figure 2.13: Test setup used for testing on normal hearing listeners [7]

Different attenuations of the processed signal were tested and results collected. The results culminated in the Articulation Index (AI) which is an objective measurement for the intelligibility of speech. AI values represent the proportion of average speech signal that is audible and range from 0 to 1 with higher values indicating greater speech intelligibility. AI scores computed from the testing can be seen in Table 2.1 below. It is apparent from these results that processing the signal has a clear improvement on speech intelligibility.

Attenuation (dB)	Processed	Unprocessed
25	0.770	0.630
30	0.636	0.490
35	0.441	0.324

Table 2.1: Articulation Index (AI) results from preliminary testing [7]

Further testing was conducted in [9] on thirty adults in the age range of 55 to 70 years that all had moderate or severe SNHL. One of the tests performed was the Revised Speech Perception in Noise Test (SPIN) which requires the listeners to repeat the last word of a sentence. Although the test was normally carried out in the presence of background noise, this test was given in quiet. Another test conducted was the Quick Speech-in-Noise Test (QSIN) which tests speech recognition in the presence of background noise. A final test conducted was the Modified Rhyme Test (MRT) which required listeners to match a word heard with the appropriate word from a list of 6 alternatives. The combination of the SPIN, QSIN and MRT experiments tested for word recognition, sentence recognition, and phoneme discrimination respectively.

Each of these tests was conducted under two signal listening conditions: unprocessed and processed with the TSEA. The order of the tests and the listening conditions was randomized to reduce order effects. Unprocessed speech samples were reduced to a sampling rate of 8000 Hz to match the sampling rate of the telephone network (and processed signals). Sample amplitudes were normalized to maximize waveform fidelity and minimize quantization errors. Results can be seen in Table 2.2 below.

	MRT %	SPIN %	QSIN %
Unprocessed			
Mean	80.5	70.3	24.8
SD	(8.6)	(15.9)	(13.9)
Processed			
Mean	85.2	81.2	56.7
SD	(7.7)	(12.9)	(21.4)

SD = standard deviation

Table 2.2: Unprocessed and TSEA processed test results [9]

This data shows that there was improvement in all three tests for the TSEA processed speech samples. However, the most significant improvement of 31.9% occurred for the QSIN test. These results further verify the effectiveness of TSEA and telephone signal preprocessing.

2.5 Additional Work Required

TSEA was developed in MATLAB and taken to the real-time Simulink environment by Kommatil [8]. It was further expanded to run as a stand-alone executable program that can run on a PC based system. Results from testing have proven that TSEA improves speech intelligibility for telephone communication.

In order for TSEA to get closer to a commercial application it must reside on an individual piece of hardware that can receive and process real-time audio speech signals. Once the speech is processed the hardware will output the signal for transmission. The following chapters explain how the algorithm was implemented on the BeagleBoard-xM hardware platform.

CHAPTER 3

THE HARDWARE DESIGN

This thesis's goal is to develop a stand-alone speech processing device that can be used for improving telephone communication. Chapter 2 described in detail the Telephone Speech Enhancement Algorithm (TSEA) that can be used to accomplish such a task. Although a stand-alone executable version of TSEA exists (Section 2.3.2), it is limited by the fact that it can only be operated via a PC environment.

A new version of TSEA is developed in this thesis that runs on the BeagleBoard-xM development board. Its implementation is outlined in this chapter as follows:

- Section 3.1 describes the BeagleBoard-xM's characteristics and capabilities.
- Section 3.2 explains how TSEA was implemented on the BeagleBoard.

3.1 The BeagleBoard-xM

When initially considering development of a hardware device to run TSEA, microcontrollers and development boards compatible with Simulink were reviewed [15]. After careful analysis the BeagleBoard-xM was chosen due to its technical specifications described in Section 3.1.1 and its Simulink compatibility described in Section 3.1.2. A picture of the BeagleBoard-xM can be seen in Figure 3.1 below.

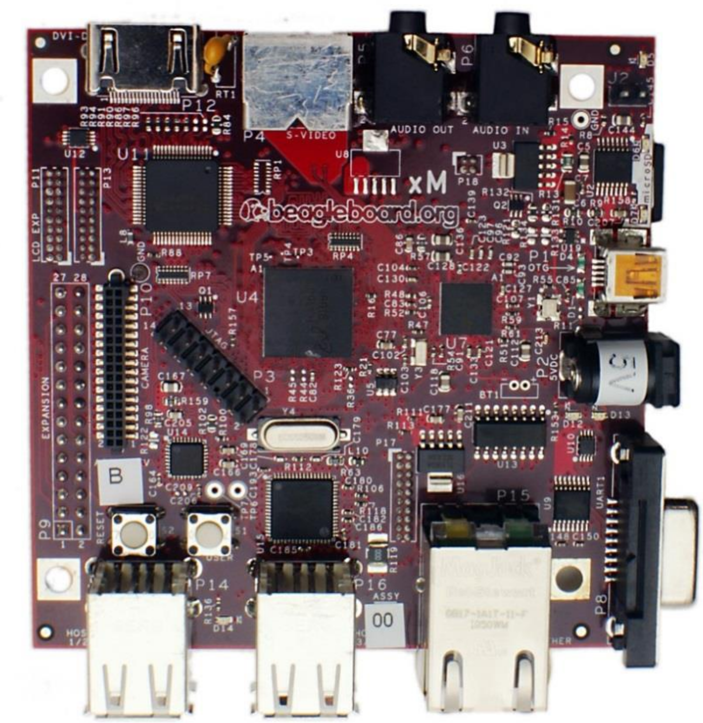

Figure 3.1: Top view of the BeagleBoard-xM [16]

3.1.1 Technical Specifications

The BeagleBoard-xM is an upgraded version of the original BeagleBoard primarily due to a higher quality processor and doubled MDDR SDRAM. Although both are powerful enough to run TSEA, the BeagleBoard-xM was chosen over the BeagleBoard because of its compatibility with Simulink. All technical specifications listed are taken from the BeagleBoard-xM system reference manual [17].

- Dimensions of the BeagleBoard-xM are 3.35” x 3.45” for a total area of 11.6 square inches. This small size is desired for the TSEA application because a commercial speech-processing hardware device must be placed near a phone where desk real estate may be limited.

- Processing for the BeagleBoard-xM comes in the form of a Texas Instruments DM3730CBP processor that can run at speeds up to 1 GHz. It is an ARM Cortex-A8 Core Digital Media Processor with a High Performance Image, Video, Audio Accelerator Subsystem [18]. This processor makes the BeagleBoard-xM more than powerful enough for running speech processing models like TSEA.
- There is 4 GBs of Micron POP (Package on Package) MDDR SDRAM which provides plenty of random-access memory for the system.
- A single microSD connector is provided on the board to allow for the main non-volatile memory storage on the board. With a decent sized microSD card (1 GB in this case) there is enough memory to store all of the programs and files needed.
- 3.5mm standard stereo audio input and output audio jack connectors are provided to allow for the connection of audio devices. This is a crucial feature for the TSEA audio processing device.
- The Texas Instruments TPS65950 Integrated Power Management IC provides power management for the BeagleBoard-xM as well as an audio codec. This codec can handle all standard audio sampling rates and has five digital-to-analog converters (DACs) and two ADCs. The input jack is connected to the auxiliary audio input of the codec and the output jack is connected to the headset differential audio output of the codec [18]. This configuration allows the BeagleBoard-xM to receive audio input signals, process them internally, and output the processed signal as desired.

- Power can be provided to the BeagleBoard-xM through a USB OTG port which can be connected to a PC that supplies power at 500mA. Another option is to use a 5V/3A DC power supply connected to the 5V power jack. This flexibility is a nice feature in case the BeagleBoard-xM needs to run in a situation where a PC USB port is not available or feasible.
- An onboard USB HUB resides on the board that uses an integrated 10/100 Ethernet which connects four USB 2.0 ports and a single Ethernet port. This Ethernet port is important in order for communication to occur between the BeagleBoard-xM and the host computer. A female DB9 serial connector is also included on the board to allow for serial communication with the host computer.
- The reset button on the board causes a power on reset when it is pushed and released. The user button is an application button that can be programmed as needed using software.

3.1.2 Simulink Compatibility

Initial requirement for a TSEA hardware platform was compatibility with Simulink. MathWorks provides extensive Simulink built-in support for the BeagleBoard-xM. This includes automated installation to help get firmware loaded onto the board. It also provides a block model library specific to the BeagleBoard-xM that contains audio input and output blocks. Interactive parameter tuning and signal monitoring gives the user further control over the system design. All of these features make the BeagleBoard-xM an excellent candidate for the TSEA implementation described in the following section.

3.2 TSEA Implementation

All of the work done by Kommatil in [8] described in Section 2.3.1 provided the foundation for the BeagleBoard-xM Simulink model. Configuring the board for use with the host computer and Simulink environment is described in Section 3.2.1. Updating the original Simulink model created several years ago for use in newer versions of Simulink is explained in Section 3.2.2. Adapting the model to run on the BeagleBoard-xM is given in Section 3.2.3.

3.2.1 Initial Configuration

Simulink support provides an automated installer for the BeagleBoard-xM which guides the user through the installation of firmware onto the board and the Simulink block library called Simulink Support Package for BeagleBoard Hardware. It is important to note that this installer is only compatible in the R2012b version of MATLAB or better. Once the installation is complete, the BeagleBoard-xM is ready to run and test compatible Simulink models.

Installation is done using the Target Installer (or Support Package Installer) which can be accessed by entering *targetinstaller* in a MATLAB command window. This opens up the installer which takes the user through a series of installation steps. First, the installer requires the user to input the desired target hardware which is the BeagleBoard in this case. It then prompts the user to connect the board to the PC via Ethernet connection and serial (COM) connection. The firmware image is then downloaded onto the PC which can take some time given that the image is approximately 1 GB. Once the firmware image download is complete it is loaded onto a 1 GB microSD card that came with the board. After being loaded, the microSD card is installed into the board which is then reset by pressing the RESET button. When the installer detects the reset via the serial port, it boots the board using the firmware image installed on the microSD card.

Now that the board has updated its firmware, it must be configured to communicate with the PC via the Ethernet port. The installer walks the user through entering a name for the board and an unused static IP address that will be used by the PC for communication. Note that the static IP address chosen must have the same subnet mask and network address as the PC. Only the host address portion of the IP address is different to distinguish the board as a unique device within the network. The Target Installer then applies these settings to the board via the serial connection. After all of this is complete, the BeagleBoard-xM is configured to run Simulink models uploaded from the PC via the Ethernet connection.

3.2.2 Simulink Model Update

Updating the old Simulink model was done by opening it in the new R2012b version of MATLAB. Initial compiling of the model resulted in errors and warnings that required debugging. Errors were generated due to new blocks created in updates that replaced the older blocks. After debugging, the Simulink model was error and warning free within the R2012b Simulink environment. This newer version of the Simulink model was compared to the older MATLAB model to verify proper functionality. A speech signal was run through both the original MATLAB model and the updated

Simulink model. Figure 3.2 shows a plot of the waveforms for the original and processed speech signals.

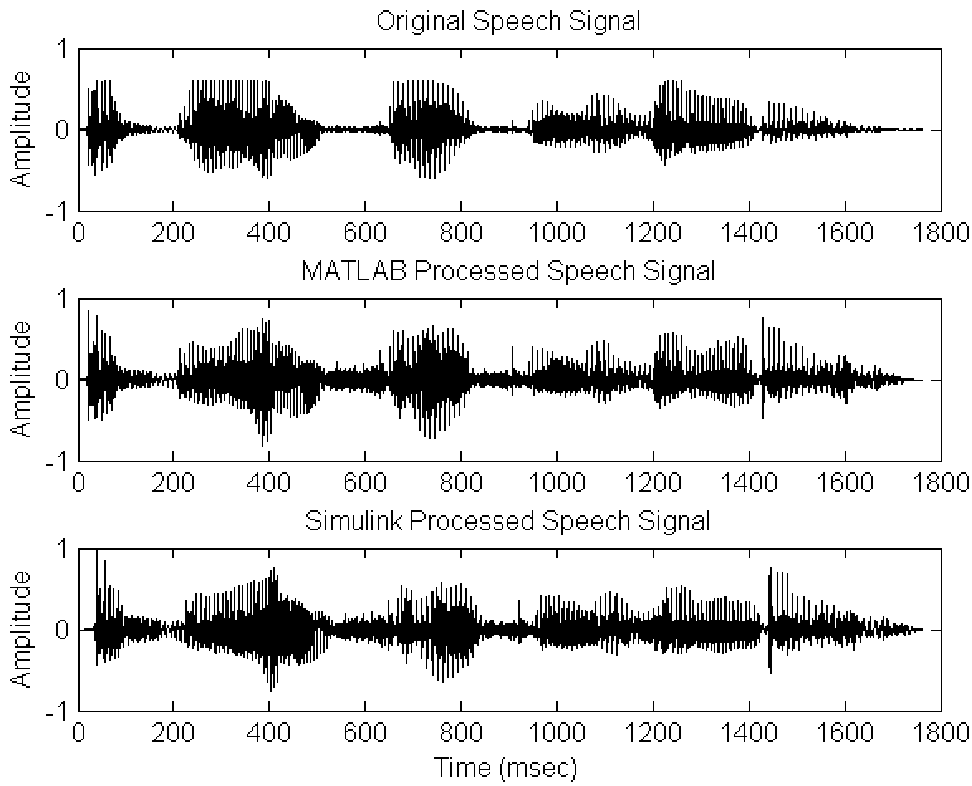

Figure 3.2: The processed speech signal from the updated Simulink model

Error computations were performed on the MATLAB processed speech signal and the Simulink processed signal in Figure 3.2. Both signals were normalized by calculating the long-term RMS voltages of the signals and scaling them to match. An RMS error of 2.12% and a maximum error of 13.51% were found for this specific input speech signal. Six additional speech signals were processed and the average RMS error was found to be 2.42% and the average maximum error was found to be 17.3%.

The delay effect mentioned in Section 2.3.1 was also observed and a close-up view of Figure 3.2 demonstrating this effect can be seen in Figure 3.3 below. Notice that the delay is still 16 milliseconds (128 samples) as shown by the difference in the first peak points. This analysis verifies that the updated Simulink model is a successful implementation of the preprocessing algorithm. This new version of the model was used as the foundation for implementation on the BeagleBoard-xM.

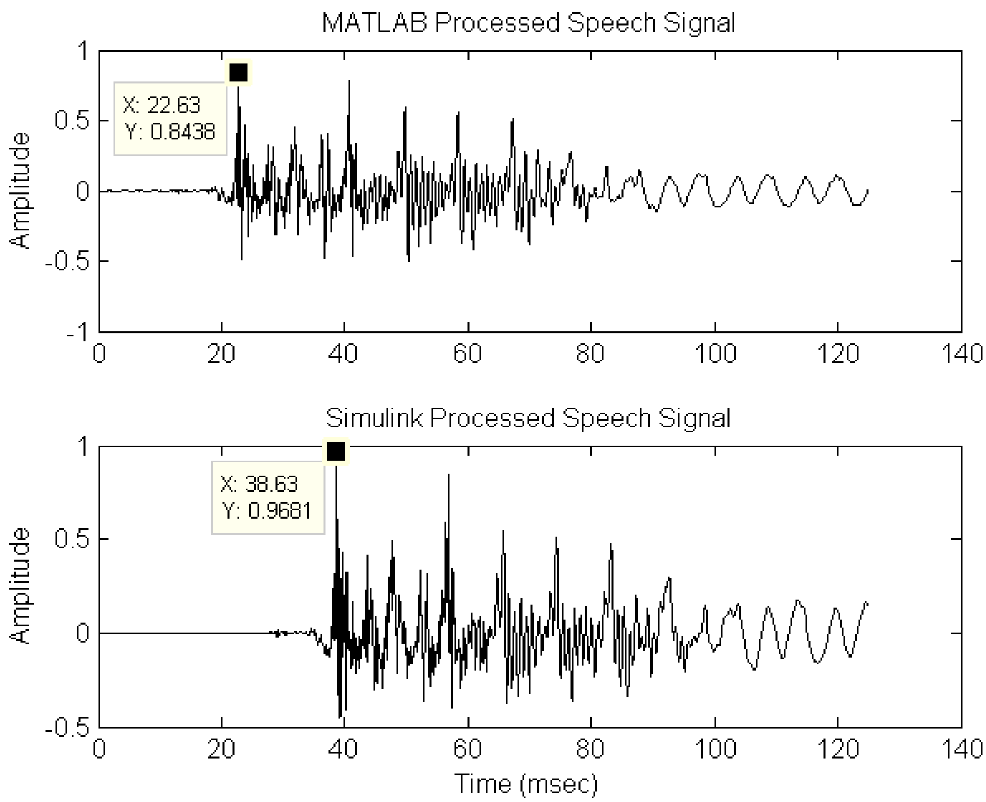

Figure 3.3: Delay effect of the updated Simulink model

3.2.3 Hardware Model Adaptation

Once the Simulink model was debugged and verified functional, it was adapted to run on the BeagleBoard-xM. Audio input and output blocks of the Simulink model had to be replaced with the blocks compatible with the target hardware. In particular, the Microphone (From Wave Device) block was replaced with the BeagleBoard ALSA Audio Capture block and the Speaker (To Wave Device) block was replaced with the BeagleBoard ALSA Audio Playback block. These new blocks were configured to run at the same 8000 Hz sampling rate with 16-bit samples and 128 samples per frame. The Audio Capture block was defined to receive 2-channel stereo signal so the model was configured to combine these signals into a single mono channel for processing. After being processed, the mono signal was then duplicated back to a 2-channel stereo signal for output by the Audio Playback block.

The To Wave File block was removed from the model because the block is platform dependent and operates only on 32-bit Windows operating systems. Unfortunately, the Simulink support package does not include an audio file read or write block. This means that the processed signal from the board can only be gathered via the audio output jack.

After the new blocks were implemented, the model was run on the BeagleBoard-xM. Changing the audio blocks created error issues with other blocks in the model. After debugging, the TSEA model was successfully uploaded to the board. With the new BeagleBoard model functioning, testing was needed to verify successful TSEA implementation as well as testing to determine the audio characteristics of the BeagleBoard-xM itself. Chapter 4 explains this process in detail.

CHAPTER 4

TESTING AND RESULTS

Testing was performed on the BeagleBoard-xM to determine its performance while running the TSEA algorithm. The BeagleBoard TSEA model was primarily compared to the Simulink model. Other tests were performed to quantify audio artifacts that the hardware itself adds to the audio signal. This chapter is laid out as follows:

- Section 4.1 shows results from the BeagleBoard to Simulink model comparison.
- Section 4.2 describes the noise characteristics added from the model.
- Section 4.3 integrates and explains the results found in Section 4.1 and 4.2.

4.1 BeagleBoard Model Testing

Speech files were sent through the BeagleBoard-xM while it was running the TSEA model. For this test, the six speech sentences used were taken from the QSIN (Quick Speech in Noise) hearing test as wave files sampled at 8000 Hz at a 16-bit depth. These parameters were used because they closely match the quality of speech signals used in telephone systems. The output of the BeagleBoard model was recorded and compared to the Simulink model.

Table 4.1 shows the error results from the comparison. These errors were computed by lining up the individual samples of the Simulink and BeagleBoard processed waveforms to match. The newly edited waveforms were then processed in MATLAB to derive the error between them. Energies of the waveforms were normalized to have equal RMS energy and the differences of the waveforms were used to calculate the RMS and maximum error.

Test	**RMS Error**	**Max Error**
Test 1	4.93%	28.68%
Test 2	4.13%	24.00%
Test 3	4.24%	37.58%
Test 4	4.38%	29.44%
Test 5	4.74%	29.48%
Test 6	3.74%	27.24%
Average	**4.36%**	**29.40%**

Table 4.1: Error between BeagleBoard and Simulink models

Error results were much higher than those found between the Simulink and MATLAB models (Section 3.2.2). Testing was performed by delivering the digital speech signals through a laptop's headphone output to the BeagleBoard's audio input jack via a 3.5mm audio cable. This emulates the telephone delivering a signal to the TSEA hardware box as shown in Figure 1.1. Once the signal was processed internally on the BeagleBoard, its output was delivered via audio cable to a computer for recording and analysis. This method introduces two digital-to-analog stages and two analog-to-digital stages into the process which will inherently degrade the speech signal's quality.

4.2 Noise Characteristics

Due to the board's analog nature (audio input/output jacks), noise enters the system and must be accounted for. Testing the noise that naturally enters the BeagleBoard-xM is described in Section 4.2.1. Checking the noise per channel is explained in Section 4.2.2.

4.2.1 Inherent Noise

It is important to know how much of the signal is degraded due to the analog nature of the system. In order to do this, sinusoidal signals at varying frequencies were sent into the BeagleBoard-xM using the same method as described in Section 4.1. The only difference was that the BeagleBoard was not running the TSEA model. Instead, the sinusoids were not processed and sent directly to the BeagleBoard's output. These sinusoid outputs were compared to the original sinusoids to determine error entering the system from the analog filtering and noise. The errors were calculated using the same method outlined in Section 4.1 and are displayed in Table 4.2.

Sinusoid Frequency	**RMS Error**	**Max Error**
100 Hz	1.34%	4.08%
500 Hz	2.87%	6.44%
1000 Hz	4.50%	8.88%
2000 Hz	2.71%	7.42%
3500 Hz	3.66%	12.02%
Average	**3.02%**	**7.77%**

Table 4.2: Error results from sinusoidal input into the BeagleBoard-xM

4.2.2 Channel Noise

Another test was done on the BeagleBoard to quantify the noise between the left and right channels of the audio codec and to determine the optimum input format. A 100 Hz sinusoid was generated in four different channel types: mono, stereo (both channels), stereo (left channel), and stereo (right channel). These were run through the BeagleBoard without processing and recorded for analysis analyzed. Results are shown in Table 4.3.

Channel Format	**RMS Error**	**Max Error**
Mono	1.61%	3.65%
Stereo	1.36%	2.98%
Left Channel	1.25%	3.67%
Right Channel	1.57%	4.28%

Table 4.3: Errors from audio channel formats

When comparing the single left channel and right channel inputs, the output from the BeagleBoard was 6 dB lower than the mono and stereo inputs. This is due to the fact that the channels are combined when converted to mono and results in constructive interference. Errors from Table 4.3 were calculated after scaling the outputs to have the same RMS energy like the previous error tests. All of these errors are small and close enough together that it can be concluded that any input format will not significantly increase or decrease performance of the BeagleBoard model.

4.3 Results Explanation

Table 4.2 results show that there is an average 3.02% error entering the system due to its analog nature and the signal path of the system. Although this partially explains the results from Table 4.1, it cannot account for the entire RMS error of 4.36% entering the BeagleBoard TSEA model. In order to account for the additional error we must examine differences between the two tests.

4.3.1 Noise Amplification

As speech signals are sent into the board, white noise spanning across the frequency spectrum is added into the system from the analog input connection. This means that both the signal and noise will be processed by the BeagleBoard TSEA model. The test in Section 4.1 processed the speech signal as well as the incoming noise introduced at the input jack. Since the noise is amplified by the algorithm explained in Chapter 2, the higher frequencies containing white noise are boosted even if there is little speech information contained in them. This phenomenon explains why the error is amplified when the TSEA algorithm is running on the model.

Figure 4.1 shows the outputs of the original waveform and the Simulink and BeagleBoard processed waveforms. From this figure, the noise entering the BeagleBoard model can be seen. Figure 4.2 shows the spectrograms of the waveforms in Figure 4.1. Again, the white noise in the BeagleBoard system can be seen as increased intensity in the spectrogram. The added noise is mostly noticeable in the higher frequency range (2-4 kHz) where greater gain is applied.

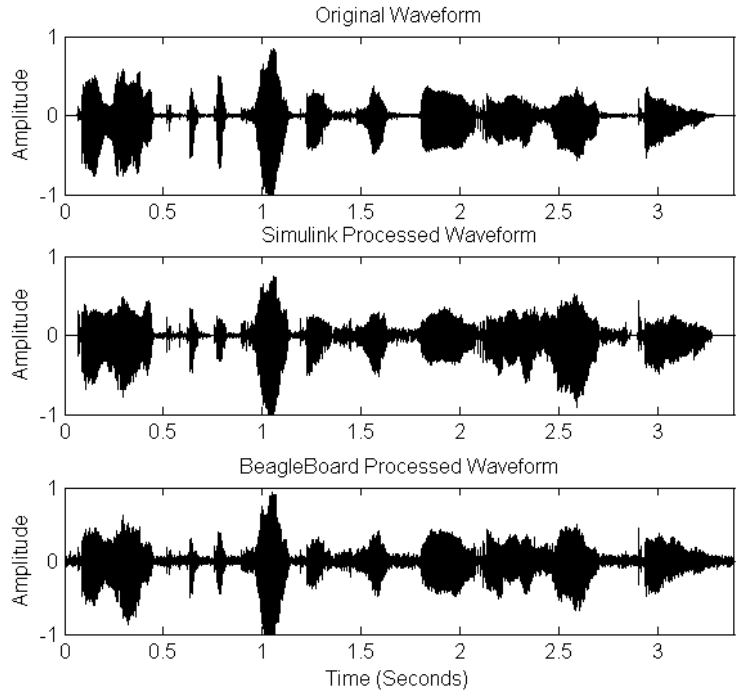

Figure 4.1: BeagleBoard processed waveform for speech sentence

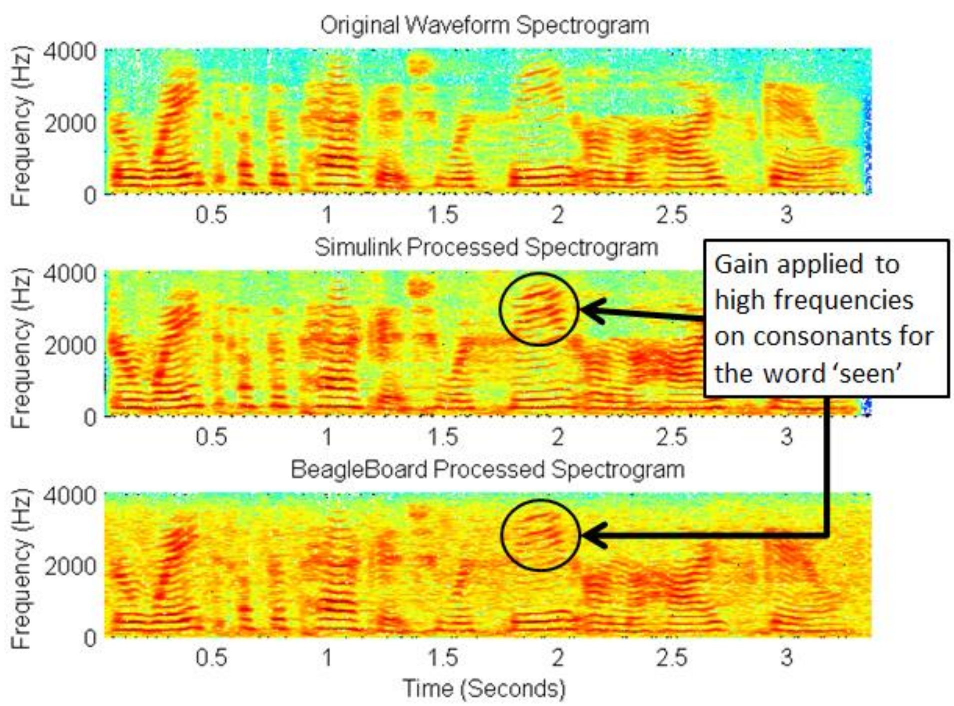

Figure 4.2: Spectrogram of waveforms in Figure 4.1

4.3.2 Noise Solutions

In order to reduce the overall noise of the BeagleBoard model's output, several strategies can be implemented. The TSEA model running on the BeagleBoard already reduces noise when there is no speech signal present. However, once the speech signal is introduced, the noise entering along with the speech signal is processed resulting in the increased error explained in Section 4.3.1.

One partial solution is to input the strongest signal into the system without clipping. Having the loudest signal possible will maximize the signal-to-noise ratio (SNR) of the input. This will result in the maximization of the SNR of the output which is desired. The only trouble with this solution is that speech varies in intensity and cannot be constantly maintained at the maximum volume constantly. Even so, making sure that the input signal is normalized to the maximum intensity will yield improved results.

Another solution to consider is incorporating a noise reduction algorithm into the model. If the BeagleBoard model is modified to recognize white noise entering the system, it could then mitigate it and process a purer speech signal. Using a low pass filter prior to sampling may also help reduce high frequency noise that folds over into the spectrum of interest. This solution would require an extension to the TSEA algorithm.

Although it is known that noise enters the BeagleBoard system, it cannot be fully determined whether or not the added noise drastically affects the performance of the TSEA itself. In order to make conclusions regarding performance with HoH listeners, human subject testing is needed. After testing is performed, results like those outlined in Section 2.4 could be tabulated to determine the model's level of effectiveness.

CHAPTER 5

CONCLUSION

5.1 Summary & Conclusions

This thesis builds upon the preprocessing algorithm, which has become known as the Telephone Speech Enhancement Algorithm (TSEA), developed by Natarajan [7] and Kommatil [8]. TSEA is implemented on the BeagleBoard-xM and can be used to operate on incoming audio signals via the audio input connector. These input signals are processed on the board and sent to the audio output connector for transmission. This model is the first hardware simulation of a potentially commercial product that can preprocess telephone signals to improve speech intelligibility for hard-of-hearing (HoH) listeners.

Results have demonstrated that while the TSEA is running on the BeagleBoard, audio artifacts enter the system in the form of noise. These artifacts have been analyzed computationally and results have shown that the amplified noise from the system is distorting the processed signal output. However, it is yet to be determined whether this noise significantly degrades the performance of the algorithm itself.

5.2 Future Work

5.2.1 Noise Reduction

Currently, the BeagleBoard TSEA model has noise entering the system that is worsened by the amplification characteristics of the algorithm. This noise can be improved by determining a way to maximize the audio input's SNR before entering the system. Ultimately, this will give the system the best conditions for achieving an optimum SNR at the output of the BeagleBoard. Also, the model itself can be further explored to determine noise reduction techniques within the algorithm. If a solution is created within the model, the noise can be mitigated and signal quality can be preserved.

5.2.2 Subject Testing

In order to determine whether the hardware model is practical for commercial application, human subject testing needs to be performed. Even with the noise entering the system, it is impossible to conclude whether or not the model improves speech intelligibility. With human subject testing, results could be gathered to decide if the model is effective enough for commercial application.

When performing human subject testing it is important to follow industry regulated standards. The standard applicable to this situation is the American National Standard Institute's (ANSI) method for measuring the intelligibility of speech over communication systems [19]. This method is known as ANSI S3.2 and was originally published in 1989 with the latest revision in 2009. It provides an outline for selecting listeners and testing them to evaluate communication systems. In this case, the communication system is the

BeagleBoard model. Testing procedures in the standard include some of the tests outlined in Section 2.4 which can be followed to make conclusions about the model's effectiveness.

5.2.3 Telephone Line Modification

Audio is sent to the board using 3.5mm audio jacks. In order for the model to fully simulate the TSEA application, the model must be able to accept telephone line signals. Peripherals could be added to the BeagleBoard-xM to allow for the connection of telephone line signals via telephone cable. Once these are added, the board could be used to send processed speech from a telephone. Receiving speech from the other end also needs to be explored since the model currently only allows for one way audio transmission.

5.2.4 Board Enclosure

Another suggestion to improve the model's design would be to create an enclosure for the BeagleBoard to reside in. The enclosure would include openings for any connections needed to be made to the board. Ultimately, it would provide the board with a layer of protection to prevent any wear and tear that may happen over time. In a commercial application this feature would be mandatory.

BIBLIOGRAPHY

[1] Wiley, T.L., Cruickshanks, K., Nondahl, D.M., Tweed, T.S., Klein, R., & Klein, B.E. “Aging and word recognition in competing message”. *Journal of the American Academy of Audiology*, 1998.

[2] Sindhusake, D., Mitchell, P., Smith, W., Golding, M., Newall, P., Hartley, D., & Rubin, G. “Validation of self-reported hearing loss. The Blue Mountains Hearing Study”. *International Journal of Epidemiology*, 2001.

[3] Popelka, M.M., Cruickshanks, K.J., Wiley, T.L., Tweed, T.S., Klein, B.E., & Klein, R. “Low prevalence of hearing aid use among older adults with hearing loss: the epidemiology of hearing loss study”. *Journal of the American Geriatric Society*, 1998.

[4] Chia, E.M., Wang, J.J., Rochtchina, E., Cumming, R.R., Newall, P., & Mitchell, P. “Hearing impairment and health-related quality of life: The Blue Mountains Hearing study”. Ear and Hearing, 2007.

[5] Kochkin, S. “MarkeTrak VI: Factors impacting consumer choice of dispenser & hearing aid brand; Use of ALDs & computers”. Hearing Review, 2002.

[6] Heine, C., & Browning, C.J. “Communication and psychosocial consequences of sensory loss in older adults: overview and rehabilitation directions”. Disability and Rehabilitation, 2002.

[7] Natarajan, Harikrishna P. “Multi-Channel Dynamic Range Compression for Telephone Speech Enhancement”. *M.S. Thesis, The Ohio State University*, 2002.

[8] Komattil, Resmi Kartha. “Simulink Implementation of a Telephone Communications Device for Elderly Individuals with Hearing Loss”. *M.S. Thesis, The Ohio State University*, 2004.

[9] Roup, CM, Poling, GL, Harhager, K, Krishnamurthy, A, & Feth, LL. (*In press*). “Evaluation of a telephone speech enhancement algorithm among older adults with hearing loss”. *Journal of Speech, Language and Hearing Research*.

[10] Krishnamurthy, Ashok. “Enhancing Speech Understanding for Audiology Telemedicine”. *Development Proposal. The Ohio State University*, 2011.

[11] Moore, Brian C.J. “Cochlear Hearing Loss”. London: Whurr. 1998.

[12] Moore, Brian C.J. “Introduction to the Psychology of Hearing”. Academic Press, Fourth Edition, 1997.

[13] Blanchard, Eugene. “Introduction to Networking and Data Communications”. Command Prompt, Inc., 2001.

[14] Sheffield, Amy E. “Telephone Speech Enhancement for Elderly Hearing-Impaired Listeners”. *M.S. Thesis, The Ohio State University,* 2000.

[15] The Mathworks. “Run Simulink Models on Supported Target Hardware”. Available at: http://www.mathworks.com/discovery/simulink-target-hardware.html, 2013.

[16] Liquidware Corporation. “BeagleBoard xM”. Available at: https://www.liquidware.com/shop/show/BB-XM/BeagleBoard+XM

[17] BeagleBoard. “BeagleBoard System Reference Manual”. Available at: http://beagleboard.org/static/BBSRM_latest.pdf

[18] Texas Instruments. “TPS65950 Integrated Power Management/Audio Codec”. Available at: http://www.ti.com/lit/ds/symlink/tps65950.pdf

[19] American National Standard. “Method for Measuring the Intelligibility of Speech over Communication Systems”. *Acoustical Society of America.* 2009.